A Late Acceptance metaheuristic for the Lock Scheduling problem

Jannes Verstichel, Greet Vanden Berghe
Vakgroep IT
KaHo Sint-Lieven
Gebroeders Desmetstraat 1
9000 Gent, Belgium

Department of Computer Science
K.U.Leuven
Celestijnenlaan 200A
3001 Leuven (Heverlee), Belgium

Email: {jannes.verstichel,greet.vandenberghe}@kahosl.be

In this abstract we present a number of metaheuristics for tackling the lock scheduling problem. The lock scheduling problem is the problem of minimising both the water usage of the lock and the waiting time of all the ships. This waiting time is the time between the arrival of a ship and its lockage time. A lock consists of at least one chamber, and its water usage is modelled by the number of lockages of the chambers. Each chamber has a limited capacity due to its size and a certain lockage time, i.e., the time needed to change the water level in the chamber from the level at one side to the level at the other side.

Efficiently handling ship operations is important due to the increasing occupation of logistic infrastructure in ports. Extensive research has been carried out on most aspects of handling ships and containers in seaports. For an overview we refer to [1]. The lock scheduling problem with multiple chamber locks is, to the best of our knowledge, new to the optimisation community.

Solving the lock scheduling problem requires a combined approach. The first part consists of solving a bin packing problem to place the ships in a chamber. The second part requires scheduling the chamber's lockages. Placing ships in a chamber is subject to four constraints (Figure 1). The size constraint (a) makes sure a ship can not be added to the chamber if it is larger than the remaining unoccupied space. A second constraints prevents overlap between ships (b). The drifting constraint (c) states that all ships must be moored during their lockage. The fourth constraint makes sure all ships are moored either to a larger ship or to the quay (d).

The performance of several local search heuristics with the best improving criterion and the late acceptance criterion is examined. These metaheuristics are multiple neighbourhood search, variable neighbourhood search and composite neighbourhood local search. The multiple neighbourhood search metaheuristic explores the neighbourhoods seperately using the same starting solution. The best solution resulting from these searches is selected as the starting solution for the next iteration. The variable neighbourhood search metaheuristic explores the neighbourhoods in a cyclic way using the resulting solution from each neighbourhood as the starting solution for the next. The composite neighbourhood search heuristic explores all the neighbourhoods at the same time.

When exploring the neighbourhoods, a candidate solution is accepted if it satisfies the *late acceptance* criterion. It depends on one parameter L only, which is the length of the acceptance list. A candidate solution's cost will be compared to the cost of the solution that was 'current' L steps before, i.e., the oldest solution in the system. When several improving moves exist, the best of them will be selected to update the acceptance list. Increasing L allows more worsening moves, and thus, to some extent, helps avoiding local optima.

Most of the neighbourhoods are generated using the corridor method. The corridor method limits the size of a neighbourhood by considering only a part of the real neighbourhood. This limitation is subject to some parameters, and promising values for these parameters are determined using a small test set. Next, we carried out experiments with all the metaheuristic optimisation methods and the best corridor parameters on a large test set. Both test sets were generated based on data from a lock of the 'Albert Kanaal' in Belgium. We will discuss the algorithms based on their performance in a set of experiments.

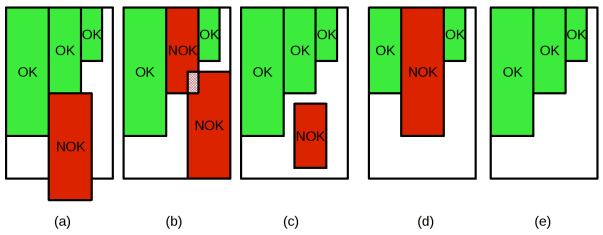

Figure 1: A visual representation of the constraints.

References

[1] R. Stahlbock, S. Voß, "Operations research at container terminals: a literature update," OR Spectrum, 2008.